

NBA Draft Day

The excitement of this year's NBA finals and draft class inspired us to make this project. With the finals coming to a close, we're soon going to be left without anything to do but sit around and wait for the next season. As a team, we love basketball so much that we didn't want to sit around but instead build some type of fantasy league for the offseason. Our problem is simple; we want a way for us, and others, to participate in the magic of the NBA during the offseason. The reason we picked the offseason is because during the season, there are plenty of fantasy leagues to participate in. However, when the offseason starts, there's nothing to do but plan for the next season. In between the end of the finals and the start of the next season, the only thing that happens for the NBA is the draft. This is where we step in. To solve our problem of having nothing to do during the offseason, we want to create an easy and fun way for people to participate in the NBA draft by being able to make their own mock drafts.

With one of the greatest NBA drafts classes coming up, we felt it was fitting to use our logic programming knowledge to build a way to participate in the fun of the NBA Draft! Our proposed solution is an interactive draft board that allows us to trade for picks and choose our favorite players to our favorite teams. Given the vast rules that trades must adhere to - team size, salary caps, and other restrictions - logic programming lends itself to ensure that these rules are not violated when making trades and picking players. Worksheets is also the ideal tool for doing this as it is very customizable and provides users with a great front-end experience powered by strong logic reasoning and rule based systems. While logic programming isn't absolutely necessary, we found it to be extremely valuable as rules of the draft and images like team logos used in the draft can be succinctly encapsulated by the combination of rulesets and HTML that comes with worksheets.

We began implementing our draft board by using the course scheduler worksheet as a base. Once the basic functionality was implemented, we began making more aesthetic choices. First of all, we needed to ensure that the list of choosing teams remains in order (unless a trade is made). We used sorted tables that would sort based off of a string predicate where the predicate would be the team's draft pick, which ensures that the teams remain in order. Some challenges we faced when doing this was that draft pick numbers are sorted as strings, and so given the numbers 1,2, 10, they would be sorted to 1,10, 2, which is not the desired effect. To remedy this, we added 100 to each number, so we then see that 101,102,110 would be properly sorted. Next, we wanted to add more images. We began by adding images of the teams: when a team is on the clock, their team logo appears on the left side of the screen. We did this using a widget from Lily and having a predicate called `selected(TEAM_NAME)` which would only be true for one team at any given time - this is the team that is currently on the clock. The draft begins with the New Orleans Pelicans as this is the first team on the clock given the actual draft order. Using this same model, we added the image of the most recently chosen player on the right side by having a separate predicate `justpicked(PPLAYER_NAME)` that is true for only one player at a time. Both

the team picture and player picture are predicates of arity 2 which essentially map the player name or team name to a url of an image. This is seen by the predicate `img(Team or player name, URL of picture)`.

In terms of functionality, we used checkboxes for each player and using precise rules ensured that they could not be deselected by the team that is choosing. We also made sure selected checkboxes could not be selected by other teams. We also added a drop down menu for the team on the clock to handle trades. After experimenting with different assumptions and methods for trading, we settled on the assumption that all teams below the current pick would want to trade up to get a better player. Once we had decided on this assumption, we made it so that the team on the clock, the team with all the power and leverage, could decide whether they wanted to trade spots in the draft with one of the lower teams which requests the trade. We made it so that it would only allow you to trade with someone who has not chosen yet and we used logic to swap the order of the two teams in the sorted table and changed their draft status from “On the clock” to “Waiting” where appropriate. Next, we added a button to allow users to clear the board and start over. As a final touch, we also added a button to allow users to share a blank sheet with other users.

Unfortunately, we were not able to add much more functionality. When working on salary caps, we came to the realization that this is too different for each team and we would need to rely on too much on data for this to work. We considered setting arbitrary salary caps for each team but felt it was too unrealistic to leave in. We spent so much time on working on salary caps that we weren't able to get very far with our other initiatives. We also attempted to include team roster limits, but once again, too much data is needed and setting an arbitrary limit felt unrealistic. For the sake of authenticity, we decided to not add arbitrary and unrealistic restrictions on trades and picks. We had also previously implemented an undo button which would allow the team that just picked to redo their pick, which entailed adjusting who is currently on the clock, deselecting the previous player, changing who the team previously selected back to “NA” and updating all of the pictures accordingly. However, NBA draft rules state that teams are not allowed a do-over; an undo button would not be realistic to the actual constraints of the NBA draft and so we removed it. Given more time, we would like to extend the project by allowing it to read from a database to enforce salary caps and team roster limits. Another addition we'd like to add is the ability to collaboratively fill in the draft board. Currently, users must gather around one computer to make choices together or make their own mock drafts where they pick for every team and compare this to the eventual draft which is what current services such as Draftkings offer.

Overall, we were very happy with our final product. We began with a lot of ambition, and while not every single goal was met, the general functionality of our final product is spot on! We are excited to use our worksheet come June 20th when we get to compare our mock drafts to the actual NBA Draft!